**Signalogic DSP Software**

# Users Guide

**(Revision 2.01)**
**September 1999**
***The one source for all of your DSP needs***

**Signalogic, Inc.**
**9617 Wendell @ Skillman**
**Dallas, Texas 75243**
**Tel: 214-343-0069**
**Fax: 214-343-0163**

The software described in this document is furnished under license agreement. The software may be used or copied only under the terms of this license agreement (see section 3.1.1).

Copyright © 1994-1999 Signalogic, Inc. All rights reserved.
No part of this manual may be photocopied or reproduced in any form without prior written consent from Signalogic, Inc.
U.S. Government: If Licensee is acquiring the software on behalf of any unit or agency of the United States Government, the following shall apply:

(a) for units of the Department of Defense:
RESTRICTED RIGHTS LEGEND: Use, duplication, or disclosure by the Government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data Clause at DFARS 252.227-7013.

(b) for any other unit or agency:
NOTICE -- Notwithstanding any other lease or license agreement that may pertain to, or accompany the delivery of, the computer software and accompanying documentation, the rights of the Government regarding use, reproduction and disclosure are as set forth in Clause 52.227-19(c)(2) of the FAR.

Contractor/mfg. is Signalogic, Inc., 9617 Wendell @ Skillman, Dallas, TX 75243.

Hyperception is a registered trademark of Hyperception, Inc.
Signalogic and DSPower are registered trademarks, and Real-Time Convolver is a trademark of Signalogic, Inc.
MATLAB is a registered trademark of The MathWorks, Inc.
National Instruments is a registered trademark, and NIDSP is a trademark of National Instruments, Inc.
Borland is a registered trademark of Borland International, Inc.
Microsoft is a registered trademark, and MS-DOS and Windows are trademarks of Microsoft Corporation.
Analog Devices, AT&T, Motorola, NEC, and Texas Instruments are registered trademarks of their respective holders. ADSP-21xx, WE DSP32x, DSP56xxx, DSP9600x, 77220, and TMS320xx are trademarks of their respective holders.
All other trademarks belong to their respective holders.

Signalogic, Inc.
9617 Wendell @ Skillman
Dallas, TX 75243
214-343-0069

Signalogic DSP Software Users Guide, June 1999
Design and Implementation: Jeff Brower.
Programming: Jeff Brower , Yolanda Lei, Ru Chen, Varsha Shamabhat.
Documentation: Jeff Brower, Varsha Shamabhat, Hanh Nguyen.

# Contents


**1 System Information 1**

Overview 1
1.1 Summary of Software Quick Reference 2
1.2 System Requirements Quick Reference 2
1.3 System Considerations Quick Reference 3
1.4 Documentation Quick Reference 3
1.5 Installation Quick Reference 4
1.6 Troubleshooting Quick Reference 5

**2 Technical Reference 7**

Overview 7
2.1 Block Quick Reference 8
2.2 Function Quick Reference 8
2.3 DSP/Analog Hardware Quick Reference 9
2.4 Waveform File Format Quick Reference 9
2.5 Definitions Quick Reference 9
2.6 Message Quick Reference 10
2.7 Error and Status Condition Quick Reference 10

**3 Using Hypersignal®-Macro Series Software 11**

Overview 11
3.1 Command-Line Interface to Hypersignal 14
3.1.1 Command-Line Options for Hypersignal 16
3.1.2 Command-Line Examples Of Running Hypersignal 21
3.2 Keyboard and Mouse Interface In Hypersignal User Mode 22
3.3 Online Help In Hypersignal User Mode 26
3.4 Main Menu in Hypersignal User Mode 27

3.4.1 Menu Selection in Hypersignal User Mode 29
3.4.2 Quit/Exit in Hypersignal User Mode 31
3.4.3 User Setup Area in Hypersignal 32
3.4.4 System Status Area in Hypersignal 35
3.5 Data-Entry Editor in Hypersignal User Mode 42
3.5.1 Waveform Summary Display in Hypersignal 46
3.5.2 Popup Directory Display in Hypersignal 48
3.6 Printing Menus and Waveform Displays in Hypersignal User Mode 52
3.7 Running External Programs in Hypersignal 53
3.7.1 DOS Shell in Hypersignal User Mode 55
3.8 Configuration Files in Hypersignal 57
3.9 Creating Waveform Files in Hypersignal 61
3.9.1 Creating Complex Waveform Files in Hypersignal 62
3.9.2 Creating Multichannel Waveform Files in Hypersignal 63
3.9.3 Creating Frequency Domain Waveform Files 64
3.10 Finding Waveforms in Hypersignal User Mode 65
3.11 Running Macro Programs in Hypersignal 70
3.11.1 Running a Macro Program From the Hypersignal Macro Manager Menu 71
3.11.2 Running a Macro Program From Outside of Hypersignal 73
3.11.3 Running a Hypersignal Macro Program from Another Macro 75
3.12 Getting Waveforms In and Out of Hypersignal User Mode 76
3.13 Displaying Time Domain Waveforms in Hypersignal User Mode 76
3.14 Interactive Waveform Displays in Hypersignal User Mode 77
3.15 FFT Generation in Hypersignal User Mode 77
3.16 Displaying Frequency Domain Waveforms in Hypersignal User Mode 77
3.17 Waveform Generation and Mathematics in Hypersignal User Mode 77
3.18 DSP/Acquisition Hardware in Hypersignal User Mode 78
3.19 Filter Design in Hypersignal User Mode 78
3.20 Applying Filters in Hypersignal User Mode 78

3.21 Creating Macro Programs and Functions in Hypersignal User Mode 78
Notes: 79

**4 Using DSPower®-HWLib and DSPower-SPLib Software 81**

Overview 81
4.1 Architecture of DSPower-HWLib and DSPower-SPLib Software 82
4.1.1 Engine and Board Handles in DSPower-HWLib and DSPower-SPLib Software 83
4.1.2 Callback Messages in DSPower-HWLib and DSPower-SPLib Software 85
4.1.3 Error and Exception Handling in DSPower-HWLib and DSPower-SPLib Software 87
4.1.4 Performance Considerations in DSPower-HWLib and DSPower-SPLib Software 91
4.2 Creating Programs for DSPower-HWLib and DSPower-SPLib 95
4.2.1 Hardware Library and Signal Processing Library Access 97
4.2.2 Include Files for DSPower-HWLib and DSPower-SPLib 98
4.2.3 .LIB Files for DSPower-HWLib and DSPower-SPLib 99
4.2.4 Example Code: DSPower-HWLib and DSPower-SPLib Software 101
4.3 DSPower-HWLib Function Overview 113
4.3.1 Hardware and Processor Control 114
4.3.2 DSP Program File Downloading 119
4.3.3 Hardware Interrogation 120
4.3.4 DSP Memory Data Transfers 126
4.3.5 Executing High-Level Functions 129
4.4 DSPower-SPLib Function Overview 130
4.5 Acquiring Data Using DSPower-HWLib Software 131

**5 Using DSP Source Code Interface Software 137**

Overview 137
5.0.1 DSP Source Code Interface Documentation 139
5.0.2 DSP Source Code Interfaces for Hypersignal 140

5.0.3 DSP Source Code Interfaces for Hypersignal in Engine Mode 147
5.1 Texas Instruments TMS320C25 Source Code Interface 148
5.1.1 Texas Instruments TMS320C25 Source Code Interface for Hypersignal 148
5.1.2 TMS320C25 Source Code Interface File List for Hypersignal 152
5.1.3 TMS320C25 Source Code Modification Process for Hypersignal 155
5.2 AT&T WE DSP32 Source Code Interface 158
5.2.1 AT&T WE DSP32 Source Code Interface for Hypersignal 158
5.2.2 DSP32 Source Code Interface File List for Hypersignal 159
5.2.3 DSP32 Source Code Modification Process for Hypersignal 160
5.3 Motorola DSP5600x Source Code Interface 163
5.3.1 Motorola DSP5600x Source Code Interface for Hypersignal 163
5.3.2 DSP5600x Source Code Interface File List for Hypersignal 167
5.3.3 DSP5600x Source Code Modification Process for Hypersignal 170
5.4 AT&T WE DSP32C Source Code Interface 173
5.4.1 AT&T WE DSP32C Source Code Interface for Hypersignal 174
5.4.2 DSP32C Source Code Interface File List for Hypersignal 177
5.4.3 DSP32C Source Code Modification Process 179
5.5 Texas Instruments TMS320C3x Source Code Interface 184
5.5.1 Texas Instruments TMS320C3x Source Code Interface for Hypersignal 185
5.5.2 TMS320C3x Source Code Interface File List for Hypersignal 188
5.5.3 TMS320C3x Source Code Modification Process for Hypersignal 191
5.6 NEC 77220 Source Code Interface 196
5.6.1 NEC 77220 Source Code Interface for Hypersignal 197

5.6.2 77220 Source Code Interface File List for Hypersignal 199
5.6.3 77220 Source Code Modification Process for Hypersignal 201
5.7 Analog Devices ADSP-21xx Source Code Interface 205
5.7.1 Analog Devices ADSP-21xx Source Code Interface for Hypersignal 205
5.7.2 ADSP-21xx Source Code Interface File List for Hypersignal 209
5.7.3 ADSP-21xx Source Code Modification Process for Hypersignal 211
5.8 Motorola DSP96002 Source Code Interface 215
5.8.1 Motorola DSP96002 Source Code Interface for Hypersignal 215
5.8.2 DSP96002 Source Code Interface File List for Hypersignal 219
5.8.3 DSP96002 Source Code Modification Process for Hypersignal 222
5.9 AT&T WE DSP3210 Source Code Interface 226
5.9.1 AT&T WE DSP3210 Source Code Interface for Hypersignal 226
5.9.2 WE DSP3210 Source Code Interface File List for Hypersignal 230
5.9.3 WE DSP3210 Source Code Modification Process for Hypersignal 232
5.10 Texas Instruments TMS320C4x Source Code Interface 236
5.10.1 Texas Instruments TMS320C4x Source Code Interface for Hypersignal 236
5.10.2 TMS320C4x Source Code Interface File List for Hypersignal 240
5.10.3 TMS320C4x Source Code Modification Process for Hypersignal 242
5.11 Texas Instruments TMS320C5x Source Code Interface 246
5.11.1 Texas Instruments TMS320C5x Source Code Interface for Hypersignal 246
5.11.2 TMS320C5x Source Code Interface File List for Hypersignal 250
5.11.3 TMS320C5x Source Code Modification Process for Hypersignal 252

## 6 Using DSPower®-Block Diagram Software 259

Overview 259
6.1 Getting Started 261
6.1.1 Installation and Checkout 263
6.1.2 Online Help in DSPower 263
6.1.3 Block Selector 264
6.1.4 Worksheet 268
6.1.5. Waveform Locator 269
6.1.6 Style/Setup Locator 271
6.2 Options in DSPower 273
6.2.1 Program Options 273
6.2.2 Worksheet Options 273
6.2.3 Diagram Options 275
6.2.4 Display Options 278
6.2.5 Help Options 279
6.3 Configuring DSPower 280
6.3.1 Directory Configuration 280
6.3.2 DSP/Analog Hardware Configuration 283
6.3.3 DSP Engine Configuration 284
6.3.4 Online Editor Configuration 286
6.3.5 Online Debugger Configuration 287
6.3.6 Block Libraries Configuration 289
6.3.7 Operator Mode (Stand-Alone Apps) Configuration 290
6.4 Creating Block Diagrams 291
6.4.1 Diagrams 291
6.4.2 Connecting Blocks 291
6.4.3 Creating Projects 291
6.4.4 Multiple Diagrams 291
6.4.5 Saving and Retrieving Block Diagrams 292
6.4.6 Executing Block Diagrams 292
6.5 Adding User Defined Blocks 292
6.5.1 Creating and Adding Blocks 292
6.5.2 Creating derivative Blocks 292
6.5.3 Adding Code Blocks 292
6.5.4 Modifying Controls 293
6.6 Interactive Operation 293
6.6.1 Run One Block 293
6.6.2 Loading Waveforms 293
6.6.3 Loading Setup and Style Files 293

6.6.4 Viewing and Editing Template and Generated Source Files 293
6.7 Tools in DSPower 294
6.7.1 Operator Mode 294
6.7.2 Make Stand-Alone Icon 294
6.7.3 Block Wizard 294
6.7.4 Waveform Headitor 294
6.8 Debugging Diagrams 294

**7 Using Engine Manager 295**

Overview 295
7.1 Engine Manager Access 296
7.2 Controlling DSP Engines 297
7.2.1 Opening DSP Engines 298
7.2.2 Sending DSP Engine Commands 300
7.3 Executing Functions in DSP Engines 301
7.4 Registering Callback Messages from DSP Engines 303
7.5 Error Handling in Engine Manager 319
7.5.1 DLL Error Handling in Engine Manager 319
7.5.2 Engine Error Handling 321
7.5.3 Error/Status Window in Engine Manager 323

**8 Using Hardware Manager 328**

Overview 328
8.1 Hardware Manager Access 329
8.2 Hardware Selector 330
8.2.1 Selecting DSP/Analog Hardware Types 331
8.2.2 DSP/Analog Hardware Configuration 332
8.2.3 Hardware Values and Variables 343
8.3 Hardware Setup File 344
8.3.1 Hardware Setup File Fields 346
8.3.2 Hardware Setup File Access 357
8.4 Error Handling in Hardware Manager 361
8.4.1 DLL Error Handling in Hardware Manager 362

**Glossary of Terms 365**

**Index 369**

# Glossary of Terms

## .ide

Filename extension used for Borland C/C++ v4.5 project files.

## .mak

Filename extension used for Microsoft Visual C/C++ v1.5 project files.

## .prj

Filename extension used for Borland C/C++ v3.1 project files.

## API

Application Programming Interface. Refers to interface presented by DLL or other Windows object to external applications.

## DDE

Dynamic Data Exchange. A Windows 3.x compatible method of transferring data between applications which don't "know" about each other. Likely to be replaced by the faster and more reliable OLE method under Windows '95.

## DLL

Dynamic Link Library. A library of routines which can be linked to any number of applications on demand, at run-time. The library is treated as an object, to which applications know how to interface, but not how it functions internally.

## DRAM

Dynamic random access memory. Typically used as large amount of auxiliary memory; for a DSP/analog board, but sometimes used instead of SRAM if the DSP device has enough onchip SRAM. 4-5 times slower and less expensive than SRAM.

## EX

Extended. Used to refer to extended functions or applications. Examples include the f_fftex and f_adex functions, and the Hypersignal-Macro EX software.

## expanded memory

Additional, usually large, amount of memory added to a PC system, but accessed through a small (64 kbyte) segment in conventional memory space. No longer as prevalent with advent of systems configured with 4 Mbyte or more of standard (extended) memory.

## extended memory

System memory located above 1024 kbyte. This is the memory referred to when a system is described as having 4 Mbyte or RAM, or 16 Mbyte of RAM, etc. installed.

## IDE

Integrated Development Environment. A term commonly used by C/C++ compiler vendors such as Borland, Microsoft, and WATCOM to refer to their collective tools and utilities when presented under a unified graphical user-interface.

## ISR

Interrupt Service Routine.  Used frequently in discussing the structure and function of DSP source code.

## MLS

Maximum Length Sequence.  A pseudo-random noise sequence of digital 1s and 0s mathematically defined to not repeat for N-1 samples, where N= 2**$^{M}$, and M is the sequence order.

## NDA

Non-Disclosure Agreement.

## NRT

Non-Real-Time.  Commonly used to refer to non-real-time versions of software.  Hypersignal-Macro NRT is an example.

## RT

Real-Time.  Commonly used to refer to real-time versions of software.

## OLE

Object Linking and Embedding.  Refers to a Windows compatible method for applications which otherwise don't "know" each other to share information and data.

## SRAM

Static random access memory.  Typically used as main program and data memory for a DSP/analog board.  4-5 times faster and more expensive than DRAM.

## VxD

Virtual Device Driver.  A type of Windows device driver used to perform low-level and/or priveleged system functions or communication.

## windows.h

Standard Microsoft Windows 3.1 include file needed by most C/C++ programs to provide common type definitions and constants..

# Index

Error! No index entries found.